



2665
EW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Dykeman et al.

Docket No.: CH919990018US1

Serial No.: 09/633,767

Group Art Unit: 2665

Filed: August 7, 2000

Confirmation No.: 3707

For: ALTERNATIVE ROUTING IN PNNI HIERARCHICAL NETWORKS

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING UNDER 37 CFR 1.8 (a)

I hereby certify that the attached correspondence comprising:


Submission of Priority Document
Certified Copy of European Patent Application No. 99115401.4
Acknowledgment Postcard

is being deposited with the United States Postal Service as first class mail in an envelope addressed to:

**Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450**

On May 24, 2005

Allison Berkman
(Type or printed name of person mailing paper or fee)


(Signature of person mailing paper or fee)

THIS PAGE BLANK (157-9)



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Dykeman et al.

Docket No.: CH919990018US1

Serial No.: 09/633,767

Group Art Unit: 2665

Filed: August 7, 2000

Confirmation No.: 3707

For: ALTERNATIVE ROUTING IN PNHI HIERARCHICAL NETWORKS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Sir:

Enclosed herewith is a certified copy of European Application No. 99115401.4
filed August 4, 1999 in support of applicant's claim to priority under 35 U.S.C. 119.

Respectfully submitted,

By

Louis P. Herzberg
Reg. No. 41,500
Phone No. (914) 945-2885

IBM Corporation
Intellectual Property Law Dept.
P.O. Box 218
Yorktown Heights, NY 10598

THIS PAGE BLANK



**Europäisches
Patentamt**

**European
Patent Office**

**Office européen
des brevets**

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

99115401.4

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 02/05/00
LA HAYE, LE

THIS PAGE BLANK (SPT0)



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

**Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation**

Anmeldung Nr.:
Application no.:
Demande n°: 99115401.4

Anmeldetag:
Date of filing: 04/08/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
International Business Machines Corporation
Armonk, N.Y. 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Alternative routing in PNNI hierarchical networks

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du dépôt:

Bemerkungen:
Remarks:
Remarques:

THIS PAGE BLANK (USPTO)

04. Aug. 1999

- 1 -

ALTERNATIVE ROUTING IN PNNI HIERARCHICAL NETWORKS

This invention relates to alternative routing of connections between source and destination nodes in PNNI hierarchical networks.

5 PNNI (Private Network-to-Network Interface) is a standards-based signaling and routing protocol which was approved by the ATM (Asynchronous Transfer Mode) Forum in 1996 for use in ATM systems. The PNNI protocol provides, inter alia, a system for creation and distribution of topology information which determines how individual network nodes "see" the network and thus how nodes communicate. A key feature of the protocol is the ability to cluster
10 groups of switches into "peer groups". The details of each peer group are abstracted into a single logical node which is all that can be seen outside of that peer group. This system is applied recursively so that PNNI can hierarchically aggregate network topology information. The hierarchical system is illustrated schematically in Figure 1 of the accompanying drawings which shows a simple, three-level hierarchical network. At the lowest level of the hierarchy in this
15 example, the nodes labeled A1 to D2 represent real switches. The switches A1, A2 and A3 are clustered to form a peer group which is represented by a logical node A in the next level up of the hierarchy. The switches B1 and B2 form a peer group which is represented by logical node B in the next level. Similarly, switches C1 to C3 and switches D1 and D2 are represented by nodes C and D respectively in the next level. Further, nodes A and B are clustered and
20 represented by logical node X in the upper level of the hierarchy. Nodes C and D are also clustered and are represented by logical node Y in the upper level.

The topology information available to the switches is such that each switch sees the details of its own peer group plus the details of any peer group that represents it at a higher level of the PNNI hierarchy. Thus, the network structure as seen by switch A1 in the above example is
25 as illustrated in Figure 2 of the accompanying drawings. Switch A1 sees two links to logical node B and one link from B to logical node Y. Node B represents all of the switches in that group, namely B1 and B2, and node Y represents all of the switches in groups C and D, namely C1, C2, C3, D1 and D2. As demonstrated by this Figure, the hierarchical aggregation of topology information leads to loss of information. For example, the details of switches C1 to D2
30 are hidden to switch A1 which sees only the logical node Y representing these switches. This

BEST AVAILABLE COPY

loss of information can lead to problems with the routing of connections for communication between nodes as will be demonstrated below.

ATM is a source routing technology. When a connection is established between nodes, the route that the connection takes through the network is specified by the first switch. This switch specifies the route as a series of paths, one for each level of the PNNI hierarchy that is traversed between the source and destination. The series of paths defining a route is known as a DTL (Designated Transit List). For example, referring again to Figures 1 and 2, suppose that a call is received at switch A1 addressed to an end system attached to switch D2. Due to the hierarchy, switch A1 does not see D2 but rather the node Y which represents D2 at a higher level. Thus, A1 may compute the following DTL for establishment of the connection:

X, Y

A, B

A1, A2

At the lowest level of the hierarchy the call goes from A1 to A2. At the second level of the hierarchy the call goes from A to B. At the top level of the hierarchy the call goes from X to the destination at Y. In the real network, the connection setup will start at A1 and be forwarded to A2 as specified. Switch A2 will forward the setup to switch B1 since it knows that its neighbor is in group B which is specified in the computed path. Switch B1 will forward the setup to switch C1 since it knows that its neighbor is in group Y which is specified in the computed path. C1 determines from the call address that the call should be forwarded to node D, and thus from C1 the call may be transferred to C2, and from there via D1 to the destination node D2.

To increase the probability that a connection can be established successfully, the ATM Forum defines a crankback mechanism. If a call setup fails, the call is returned to node(s) that created the path to attempt to choose an alternative path ("alternative routing"). To achieve this, the call is returned with a crankback element giving an indication as to which element of the network failed to forward the call. To illustrate the crankback mechanism, consider the situation where, in establishing the connection in the above example, the links from C1 to C3 and from C1 to C2 do not have sufficient bandwidth to support the call. In this situation the call cannot be established successfully since C1 has no way to forward it to the destination D2. The crankback mechanism is therefore used to return the call for possible alternative routing. The call is returned by C1 to B1. Since the lowest common level of hierarchy between these two nodes is

BEST AVAILABLE COPY

BEST AVAILABLE COPY

that containing nodes X and Y, the call is returned with cause code "Node Y is blocked." C1 cannot tell B1 about the internal links and switches in its peer group since B1 cannot see them. Since node B did not specify the path X, Y it will not attempt to reroute the call. The call is next returned to switch A2, but since A2 did not specify the path, it is immediately returned to switch

5 A1. Switch A1 is thus informed that the call setup failed because node Y is blocked. However, in the network structure as seen by A1, node Y constitutes an element which provides the only way to access the destination node D1 to which the call must be sent. Such an element, which provides sole access to a destination node, will be referred to herein as a "sole-access element". According to the rerouting scheme which is specified in the PNNI protocol, switch A1 will not

10 try to find an alternative route for the call since a sole-access element, here node Y, is blocked. Thus, a successful connection cannot be established and the call setup fails.

The weakness of the above system can be seen from consideration of Figure 1 which shows that there is a route over which the call could be established successfully, namely: A1, A3, B2, C3, C2, D1 and D2. Thus, if switch A1 had originally specified the route:

15 X, Y
A, B
A1, A3

then the call setup would have been successful. However, even with the crankback mechanism, the alternative routing scheme fails because a significant part of the network topology is hidden

20 from the switch attempting to reroute the connection.

According to one aspect of the present invention there is provided a method for alternative routing of a connection between a source node and a destination node in a PNNI hierarchical network, the method comprising responding to a failed connection between said nodes due to a sole-access element of the network structure as seen by the source node, where a

25 said sole-access element is an element which provides sole access to the destination node in said network structure, by:

selecting at least one non-sole-access element of the route used by the failed connection in said network structure;

identifying an alternative route for the connection in said network structure which does

30 not utilize the at least one selected element; and

using the alternative route for establishment of the connection between said nodes.

The present invention is predicated on the realization that a failed connection due to a sole-access element of the network structure seen by the source node can occur due to selection of another element in the route. Thus, by altering the selection of one or more non-sole-access elements (ie. elements which are not sole-access elements), a route by which the connection can be established correctly can often be found. For example, in the scenario discussed above where the connection failed on the basis that the sole-access node Y was blocked, a different result could have been achieved by making a different selection of the non-sole access elements earlier in the route. Evidently if there are no non-sole access elements in a particular failed route then there will be no alternative route to try. However, where there is at least one non-sole access element there will be at least one alternative route, and embodiments of the present invention can identify this and reroute the connection accordingly. Thus, embodiments of the present invention allow successful rerouting of connections where the previous rerouting scheme would have failed, improving the overall performance and robustness of networks in which the invention is applied.

In preferred embodiments of the invention, the method includes the step of checking whether the alternative route satisfies a set of predefined connection constraints, wherein the alternative route is used for establishment of the connection only if said constraints are satisfied. The set of connection constraints may include one or more constraints and these may vary for different connections. In particular, PNNI supports the use of Quality of Service (QoS) parameters whereby incoming calls can specify certain constraints which the network must satisfy when establishing the connection. Typical constraints relate, for example, to the bandwidth required by a call and the acceptable delay introduced by the connection. The PNNI protocol allows a source node to determine if these constraints can be met by a particular route, and if not the route will be rejected.

While methods can be envisaged in which the selected element or elements of the failed route are nodes of the network structure, it is preferred that the selected elements, ie. the elements which are avoided when determining the alternative route, are links of the network structure. Various systems can be used for selecting the particular elements to be avoided when determining an alternative route, and examples of preferred systems will be described in detail below.

In methods embodying the invention, if a successful connection is not established with a first-identified alternative route (eg. because the route does not meet any specified connection constraints or because the connection fails again when the alternative route is used), then the method may be applied again making a different selection of non-sole-access elements if available, to identify a different alternative route. Thus, methods embodying the invention may be iterative. The number of attempts which are made to find a successful route may of course be limited to avoid excessive use of resources. For example, while multiple attempts may be made to identify a "valid route" (ie. a route which meets any specified connection constraints), the number of attempts to reroute a connection using different valid routes may be limited. In particular, some embodiments may make only one attempt at rerouting a connection in order to preserve network resources.

It is to be understood that, in general, where features are described herein with reference to a method embodying the invention, corresponding features may be provided in accordance with apparatus embodying the invention, and vice versa. In particular, a further aspect of the present invention provides apparatus for alternative routing of a connection between a source node and a destination node in a PNNI hierarchical network, the apparatus comprising:

memory for storing topology data, defining the network structure as seen by the source node, and route data indicative of a route in said network structure used for establishment of a connection between the source node and a destination node;

control logic configured to respond to a failed connection between said nodes due to a sole-access element of the network structure as seen by the source node, where a said sole-access element is an element which provides sole access to the destination node in said network structure, by:

selecting at least one non-sole-access element of the route used by the failed connection in accordance with said route data;

identifying from said topology data an alternative route for the connection which does not utilize the at least one selected element; and

outputting the alternative route for establishment of the connection between said nodes.

The apparatus may be embodied in a source node such as a switch, router, bridge, brouter or other network device. Alternatively, the apparatus may be embodied in a dedicated route

server associated with a peer group including the source node. The invention also extends to a PNNI hierarchical network comprising such apparatus.

Preferred embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings in which:

- 5 Figure 1 is a schematic representation of an example of a PNNI hierarchical network;
 Figure 2 illustrates the network structure as seen by the switch A1 in Figure 1;
 Figure 3 is a schematic representation of another example of a PNNI hierarchical network; and

 Figure 4 illustrates the network structure as seen by the switch A1 in Figure 3.

- 10 In the following, a preferred method embodying the invention will initially be described in detail, and operation of the preferred method will be described in relation to the networks of Figures 1 and 3. For the purpose of this example, it will be assumed that each switch in the exemplary networks is capable of implementing the rerouting method, so that each switch in its own right constitutes apparatus embodying the invention. To enable a switch to implement the
15 alternative routing method, control logic for implementing the various steps to be described is provided in the switch. The control logic may be implemented in hardware or software or a combination thereof, and suitable implementations will be apparent to those skilled in the art from the following description.

- The rerouting method performed by a switch can be summarized conveniently by the
20 following pseudo-code, where it is assumed initially that a failed connection between that switch and a destination node has just occurred.

1. Block all the links used by the failed connection that are outside our peer group;
- 25 2. if (any link is a sole-access link)
 then do
 unblock the link giving access to it;
 done

- 7 -

3. Store the list of blocked links in BL;
Sort BL by decreasing distance from source node (links closest to the destination at the top);
- 5 4. Compute the route;
if (route is valid)
 then do
 give the route to signalling;
 End;
10 done
5. Unblock all the links;
6. for all the elements in BL do
15 Unblock previous element if there was one;
 Block the current element;
 Compute the route;
 if (route is valid)
 then do
20 Give the route to signalling;
 End;
 done
 done
25 7. Give 'No route to destination' to signalling.

BEST AVAILABLE COPY

In the above algorithm, a number of the steps refer to blocking or unblocking of certain links. In reality, the links are not of course physically blocked or unblocked, but rather certain links are selected or deselected, and as a result of this selection/deselection process certain links will be considered to be blocked for the purpose of identifying an alternative route. This will be

apparent from the following description in which the way that the algorithm is implemented by a source switch is considered in more detail.

In step 1, the switch control logic initially selects ("blocks") all links of the route used by the failed connection which are outside its own peer group. The links used by the failed
5 connection are defined by route data which has previously been stored in a memory of the switch and can be accessed by the control logic when required for the alternative routing scheme. In step 2, the control logic deselects ("unblocks") any sole-access links since these must be used by any route to the destination node. The control logic can identify sole-access links by reference to the topology data which has been set up previously in accordance with the PNNI protocol and
10 which is stored in the switch memory. This topology data defines the network structure seen by the switch as described earlier. In step 3, the control logic stores the remaining selected links, (the "blocked links"), as a blocked-link list BL in the switch memory. The control logic then sorts the links in BL in order of decreasing distance from the source node such that the link which is closest to the destination node is at the top of the list.

15 After this initial selection process, in step 4 the control logic analyzes the topology data with reference to the list BL to determine if there is an alternative route to the destination which does not utilize any of the "blocked" links in BL. If an alternative route is identified, the control logic then checks, by means of the known processing operations specified by PNNI which need not be detailed here, whether the route satisfies any connection constraints, such as QoS
20 bandwidth or delay requirements etc., which have been specified by the incoming call. Again, these constraints will be stored in the switch memory for access by the control logic. If the constraints are met then the alternative route is deemed to be valid. If the route is valid, the control logic outputs the route to the signaling circuitry of the switch (not described but of known form) for establishing the connection, and the process is finished. If, on the other hand an
25 alternative route identified in this step does not satisfy the connection constraints and is therefore determined to be invalid, the process proceeds to step 5 wherein all the links are "unblocked". Again, this "unblocking" of the links in step 5 is only notional in that the "blocked" links in BL are simply treated by the control logic as deselected, or "unblocked", for the start of step 6 to which the process now proceeds.

30 Step 6 begins with an iterative process in which the control logic selects, or "blocks", each link in BL in turn, starting at the top of the list. When a link is selected, the control logic

then determines as described above whether there is a valid alternative route which does not utilize the selected link. If so the route is output to the signaling circuitry for rerouting the connection and the process terminates. If there is no valid route for the currently-selected link then the process proceeds to the next pass of the iteration in which the currently-selected link is deselected, the next link in the list BL is selected, and a valid alternative route is sought which does not utilize the newly-selected link. If, after all links in BL have been selected in turn, no valid route has been found, then the process proceeds to step 7 wherein the control logic outputs a message indicating that no valid route to the destination node is available. This message is communicated to the signaling circuitry for transfer to the device which originated the call.

Consideration of the above rerouting algorithm shows that it can be divided into two parts. The first part consists of steps 1 to 4. This part aims to identify the most disjoint (most different) route to that used by the failed connection. This is done by selecting all links in the set of non-sole-access links in the route used by the failed connection which are outside the peer group of the source switch, and identifying an alternative route which does not utilize any of these links. If this fails then the second part, consisting of steps 5 to 7, is implemented. This part involves selecting the non-sole access links of the aforementioned set one at a time, starting with the link closest to the destination node, and checking for an alternative route which does not utilize the selected link.

This particular algorithm makes only one attempt at rerouting a failed connection. Thus, when a valid alternative route has been used for establishing a connection, if the connection fails again then no attempt is made to find another alternative route. The first part of the method looks for the most disjoint route since this generally has the best chance of success if used for rerouting the connection. In the second part of the method, selecting links in order of increasing distance from the destination node makes it likely that a more disjoint path, which generally provides a better chance of a successful connection than a less disjoint path, will be found first. Thus, while only one rerouting attempt is made to keep use of network resources to a minimum, the method ensures that there is a good chance that this one attempt will be successful.

To demonstrate operation of the rerouting method, consider first the situation previously described with reference to Figure 1 where a failed connection between source node A1 and destination node D2 has just occurred due to insufficient bandwidth in the links from C1 to C2 and C1 to C3 with the following route:

- 10 -

X, Y

A, B

A1, A2

In this case, switch A1 is a switch embodying the invention which can implement the preferred method as described above. Thus, when the call is returned to A1 with the cause code "Node Y is blocked", the rerouting algorithm is implemented even though node Y is a sole-access element. In accordance with step 1 of the method, A1 first "blocks" all links used by the failed connection outside of its peer group, namely A2 - B, and B - Y. It is apparent from Figure 2, however, that the link B - Y is a sole-access link so this link is "unblocked" in step 2. The list BL in step 3 therefore contains only A2 - B. The alternative route identified in step 4 is therefore:

X, Y

A, B

A1, A3

15 Assuming this route satisfies any specified connection constraints and is therefore deemed valid, the route will be output for establishment of the connection. In the real network, the route will cause the connection to go through the following sequence of switches: A1, A3, B2, C3, C2, D1 and D2, and the connection will be established successfully.

Consider next the two-level hierarchical network of Figure 3. The lowest level of the hierarchy in this network is similar to that of Figure 1 except that there are additional switches C4 and D3 interconnected as shown. The four groups of switches A1 to A3, B1 and B2, C1 to C4, and D1 to D3 are clustered to form peer groups which are represented by logical nodes A, B, C and D in the upper level of the hierarchy as indicated. In this example, however, the links interconnecting peer groups are not aggregated in the hierarchy, so that, in the upper level, successive nodes A, B, C and D are interconnected by two links. In each pair of links interconnecting two nodes, the upper link in each pair is labeled 1 in the figure indicating that this link originates at a first port (port 1) of the preceding node. The lower link of each pair is labeled 2 in the figure indicating that this link originates at a second port (port 2) of the preceding node. The network structure as seen by node A1 in this case is as shown in Figure 4.

30 Suppose that, as before, switch A1 has received a call addressed to an end system attached to D2 and computes the following initial route for the connection:

- 11 -

A1, B1, C1, D

A1, A2

where, in the path A1, B1, C1, D, the notation "A1" signifies node A, port 1, "B1" signifies node B, port 1, and so on, so that the upper links 1 in Figure 4 should be used. In the real network the call setup is forwarded by A1 to A2 as specified in the route. Switch A2 then forwards the setup directly to B1 via the upper link 1 as specified in the route. Similarly, B1 forwards the setup directly to C1 via the upper link 1 as specified in the route. Suppose that, as before, the links from C1 to C2 and from C1 to C3 do not have sufficient bandwidth to support the call so that the call cannot be completed. The crankback mechanism is then used to return the call to the source switch A1 with the cause code "Node C is blocked". Node C is a sole-access element of the network structure as seen by A1 since any connection to the destination node for the call must utilize node C. Thus, with the existing rerouting scheme specified by PNNI, A1 would not attempt to reroute the call since a sole-access element is blocked. Here, however, the switches A1 to D2 are configured for implementing the rerouting algorithm described in detail above. Thus, when the call is returned to A1, A1 will initially block the links A1 - B, B1 - C and C1 - D used by the failed connection outside its peer group (step 1). None of these links are sole-access links (step 2), so that, at the end of step 3, the blocked link list BL will contain all these links in the order C1 - D, B1 - C, A1 - B. In step 4, A1 computes the following route which uses none of the "blocked" links in B:

A2, B2, C2, D

A1, A3

where, in the path A2, B2, C2, D, the notation "A2" signifies node A, port 2, "B2" signifies node B, port 2, and so on, so that the lower links 2 in Figure 4 should be used. For the sake of example, assume now that this route does not meet the specified delay constraints for the call, so that the route is not valid. All the links in BL are then "unblocked" (step 5) and the process proceeds to step 6. In the first stage of step 6, the first link in the list BL, ie. C1 - D is initially blocked and the control logic of A1 computes an alternative route which does not utilize this link. Since there is more than one possible route here, in this embodiment the control logic selects a particular route at random, though of course other selection systems could be used if desired. Thus, for example, A1 may identify the following route:

BEST AVAILABLE COPY

- 12 -

A1, B2, C2, D

A1, A2

Again for the sake of example, suppose that this route is deemed invalid. Switch A1 will then continue the search for a valid route by selecting another route which avoids the "blocked" link

- 5 C1 - D, for example using the upper-level path A2, B1, C2, D. Suppose further that this route, and each of the other possible routes which do not use the link C1 - D, is also found to be invalid. Switch A1 then continues with step 6 by unblocking C1 - D and blocking the next link in BL, namely B1 - C. A1 now computes a route which does not utilize B1 - C. Again, since there is more than one route to choose from here, a particular route will be selected at random.
- 10 This may be a route already identified as an invalid route, eg. the route using the upper level path A1, B2, C2, D, so the control logic will select another route, eg:

A2, B2, C1, D

A1, A3.

Assuming this route satisfies the connection constraints, the route will be output to signaling. In

- 15 the real network, the connection will pass through the following sequence of switches: A1, A3, B2, C3, C2, D1 and D2, and the connection will be established successfully.

When considering the application of the preferred algorithm in the networks of Figures 1 and 3, the description has focused on examples of failed connections where the failure is due to a sole-access element of the network structure as seen by the source node since these are the

- 20 situations where the existing rerouting scheme would inevitably fail to find an alternative route. It will be appreciated however, that in practice the algorithm will be applied in response to any failed connection whether or not due to a sole-access element, and provides a highly efficient rerouting system in all circumstances.

It will also be appreciated that, while a particularly preferred embodiment has been

- 25 described in detail above, many changes and modifications can be made to the embodiment described without departing from the scope of the invention. For example, in some situations it may be acceptable, or even preferable, for step 3 of the algorithm to sort the links in BL by increasing distance from the source node, ie links closest to the source node at the top. In some embodiments, therefore, the control logic may be configured to implement the algorithm with
- 30 this modification in all cases, or the control logic may be configured to select which of the two orders to adopt in a particular situation. In addition, in some embodiments step 6 of the above

- 13 -

algorithm may be modified by removal of the statement "unblock previous element if there was one". The effect of this would be that, when performing step 6, the control logic would successively block more links in each pass of the iterative process. This would reduce processing time by restricting the number of routes tried by the control logic, while still providing a good
5 chance of successfully rerouting a connection.

As a further example, some embodiments may utilize only one part of the algorithm described above, ie. steps 1 to 4 or steps 5 and 6. While the two-part algorithm is clearly more likely to result in successful rerouting, use of either part on its own will still allow successful rerouting to be achieved in many cases where the prior scheme would have failed, and these
10 simpler algorithms may be appropriate in some circumstances, eg. where it is desired to simplify the processing as far as possible.

Further, while the above method makes only one attempt at rerouting a connection, in some embodiments it may be desirable to make more than one attempt, or even to try all possible valid routes before returning a "No route to destination" code. In these cases, if a valid
15 alternative route is used for establishing a connection and the connection fails again, then the failed route may be stored in the control logic as an invalid route, and the method repeated to identify a new alternative route.

As a further example, in the networks as described above, each switch incorporates control logic for implementing the rerouting method when a failed call is returned to the switch.
20 In other embodiments, the rerouting method may be implemented by a dedicated node, such as a route server, which manages rerouting of calls for any source node within a particular peer group. Thus, one such route server may be provided for each peer group, the route server being connected in the peer group and sharing the same view of the network topology as the other nodes in the group. When a failed call is returned to a source node in a given peer group, the
25 source node simply sends the details of the failed route to the route server together with any connection constraints and the cause code for the connection failure. The route server then implements the rerouting method as described above, returning the alternative route to the source node for forwarding to signaling.

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

04. Aug. 1999

CLAIMS

1. A method for alternative routing of a connection between a source node and a destination node in a PNNI hierarchical network, the method comprising responding to a failed connection
5 between said nodes due to a sole-access element of the network structure as seen by the source node, where a said sole-access element is an element which provides sole access to the destination node in said network structure, by:
selecting at least one non-sole-access element of the route used by the failed connection in said network structure;
10 identifying an alternative route for the connection in said network structure which does not utilize the at least one selected element; and
using the alternative route for establishment of the connection between said nodes.
2. A method as claimed in claim 1 including checking whether the alternative route satisfies
15 a set of predefined connection constraints, wherein the alternative route is used for establishment of the connection only if said constraints are satisfied.
3. A method as claimed in claim 1 or claim 2 wherein said at least one element is a link of said network structure.
20
4. A method as claimed in claim 3 including selecting all non-sole-access links of the route used by the failed connection which are outside the PNNI peer group of the source node.
5. A method as claimed in claim 3 including selecting from the set of all non-sole-access
25 links used by the failed connection which are outside the PNNI peer group of the source node the link which is closest to a predetermined one of the source and destination nodes.
6. A method as claimed in claim 5 including:
(a) if a successful connection is not established using the alternative route, selecting from
30 said set of non-sole-access links the next closest link to the predetermined node, identifying a

new alternative route for said connection which does not utilize said next closest link, and using the new alternative route for establishment of the connection between said nodes; and

(b) repeating step (a) for the new alternative route until all links in said set have been selected.

5

7. A method as claimed in claim 6 wherein the new alternative route does not utilize any link of said set between said predetermined node and said next closest link.

8. A method as claimed in claim 4 wherein, if a successful connection is not established
10 using said alternative route, the method includes:

(a) selecting from the set of all non-sole-access links used by said failed connection which are outside the PNNI peer group of the source node the link which is closest to a predetermined one of the source and destination nodes, identifying a new alternative route for the connection which does not utilize said closest link, and using the new alternative route for
15 establishment of the connection between said nodes; and

(b) if a successful connection is not established using the new alternative route, selecting from said set of non-sole-access links the next closest link to the predetermined node, identifying a new alternative route for said connection which does not utilize said next closest link and using the new alternative route so identified for establishment of the connection between said nodes;
20 and

(c) repeating step (b) for the new alternative route so identified until all links in said set have been selected.

9. A method as claimed in claim 8 wherein the new alternative route identified in step (b)
25 does not utilize any link of said set between said predetermined node and said next closest link.

10. A method as claimed in any one of claims 6 to 9 including checking whether an identified new alternative route satisfies a set of predefined connection constraints, wherein the identified new alternative route is used for establishment of the connection only if said
30 constraints are satisfied.

11. A method as claimed in any one of claims 5 to 10 wherein said predetermined node is the destination node.
- 5 12. Apparatus for alternative routing of a connection between a source node and a destination node in a PNNI hierarchical network, the apparatus comprising:
- memory for storing topology data, defining the network structure as seen by the source node, and route data indicative of the route in said network structure used for establishment of a connection between the source node and a destination node;
- 10 control logic configured to respond to a failed connection between said nodes due to a sole-access element of the network structure as seen by the source node, where a said sole-access element is an element which provides sole access to the destination node in said network structure, by:
- selecting at least one non-sole-access element of the route used by the failed connection
- 15 in accordance with said route data;
- identifying from said topology data an alternative route for the connection which does not utilize the at least one selected element; and
- outputting the alternative route for establishment of the connection between said nodes.
- 20 13. Apparatus as claimed in claim 12 wherein the control logic is configured to check whether the alternative route satisfies a set of predefined connection constraints, and to output the alternative route for establishment of the connection only if said constraints are satisfied.
14. Apparatus as claimed in claim 12 or claim 13 wherein said at least one element is a link
- 25 of said network structure.
15. Apparatus as claimed in claim 14 wherein the control logic is configured to select all non-sole-access links of the route used by the failed connection which are outside the PNNI peer group of the source node.

16. Apparatus as claimed in claim 14 wherein the control logic is configured to select from the set of all non-sole-access links used by the failed connection which are outside the PNNI peer group of the source node the link which is closest to a predetermined one of the source and destination nodes.

5

17. Apparatus as claimed in claim 16 wherein the control logic is configured such that:

(a) if a successful connection is not established using the alternative route, the control logic selects from said set of non-sole-access links the next closest link to the predetermined node, identifies a new alternative route for said connection which does not utilize said next

10 closest link, and outputs the new alternative route for establishment of the connection between said nodes; and

(b) the control logic repeats step (a) for the new alternative route until all links in said set have been selected.

15 18. Apparatus as claimed in claim 17 wherein the new alternative route does not utilize any link of said set between said predetermined node and said next closest link.

19. Apparatus as claimed in claim 15 wherein the control logic is configured such that, if a successful connection is not established using said alternative route:

20 (a) the control logic selects from the set of all non-sole-access links used by said failed connection which are outside the PNNI peer group of the source node the link which is closest to a predetermined one of the source and destination nodes, identifies a new alternative route for the connection which does not utilize said closest link, and outputs the new alternative route for establishment of the connection between said nodes; and

25 (b) if a successful connection is not established using the new alternative route, the control logic selects from said set of non-sole-access links the next closest link to the predetermined node, identifies a new alternative route for said connection which does not utilize said next closest link, and outputs the new alternative route so identified for establishment of the connection between said nodes; and

(c) the control logic repeats step (b) for the new alternative route so identified until all links in said set have been selected.

20. Apparatus as claimed in claim 19 wherein the new alternative route identified in step (b) does not utilize any link of said set between said predetermined node and said next closest link.
21. Apparatus as claimed in any one of claims 17 to 20 wherein the control logic is configured to check whether an identified new alternative route satisfies a set of predefined connection constraints, and to output the identified new alternative route for establishment of the connection only if said constraints are satisfied.
22. Apparatus as claimed in any one of claims 16 to 21 wherein said predetermined node is the destination node.
23. A source node of a PNNI hierarchical network, the source node comprising apparatus as claimed in any one of claims 12 to 22 for alternative routing of a connection between that source node and a destination node in the network.
24. A route server for association with a peer group of nodes in a PNNI hierarchical network, the route server comprising apparatus as claimed in any one of claims 12 to 22 for alternative routing of a connection between a source node in said peer group and a destination node in the network.
25. A PNNI hierarchical network comprising apparatus as claimed in any one of claims 12 to 22.

THIS PAGE BLANK (USPTO)

04. Aug. 1999

ABSTRACTALTERNATIVE ROUTING IN PNNI HIERARCHICAL NETWORKS

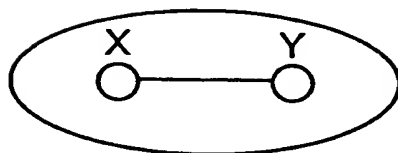
Methods and apparatus are provided for alternative routing of a connection between a
5 source node and a destination node in a PNNI hierarchical network. The alternative routing
method comprises responding to a failed connection between said nodes due to a sole-access
element of the network structure as seen by the source node, where a said sole-access element is
an element which provides sole access to the destination node in said network structure, by:
selecting at least one non-sole-access element of the route used by the failed connection in said
10 network structure; identifying an alternative route for the connection in said network structure
which does not utilize the at least one selected element; and using the alternative route for
establishment of the connection between said nodes.

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

BEST AVAILABLE COPY

1/2



EPO-Munich
52

04. Aug. 1999

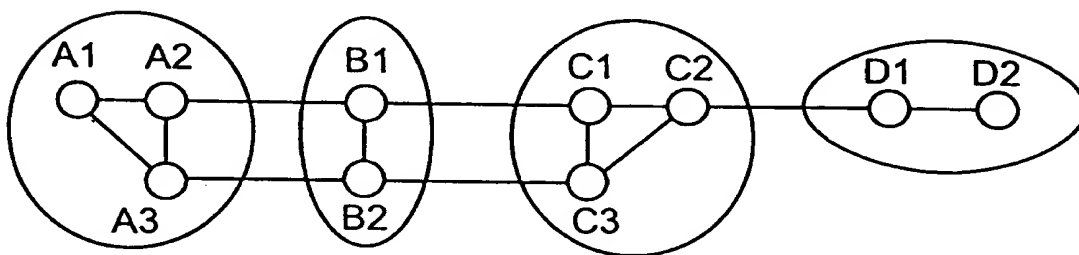
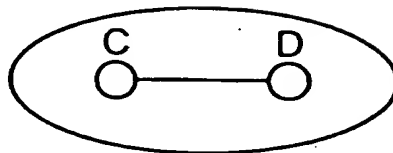
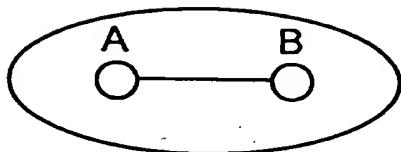


Figure 1

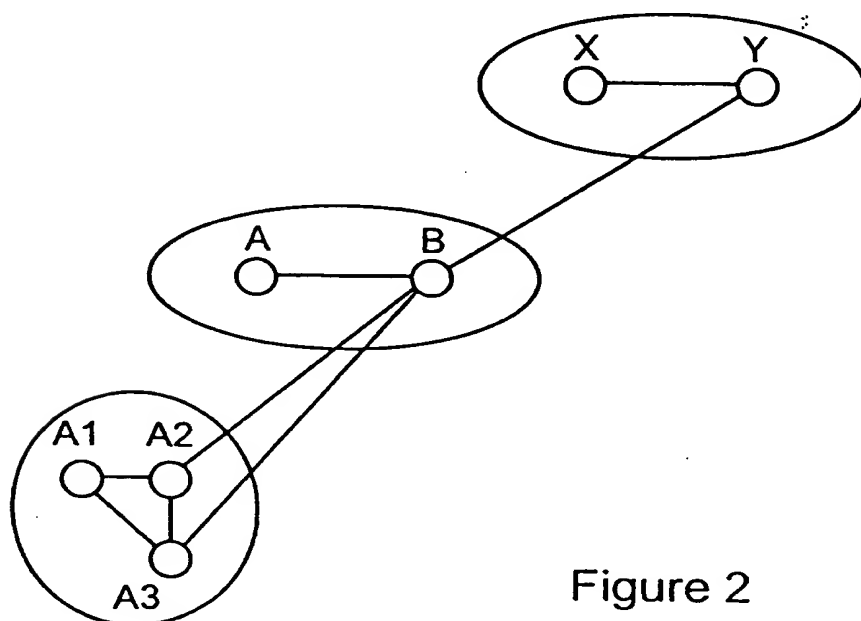


Figure 2

BEST AVAILABLE COPY

2/2

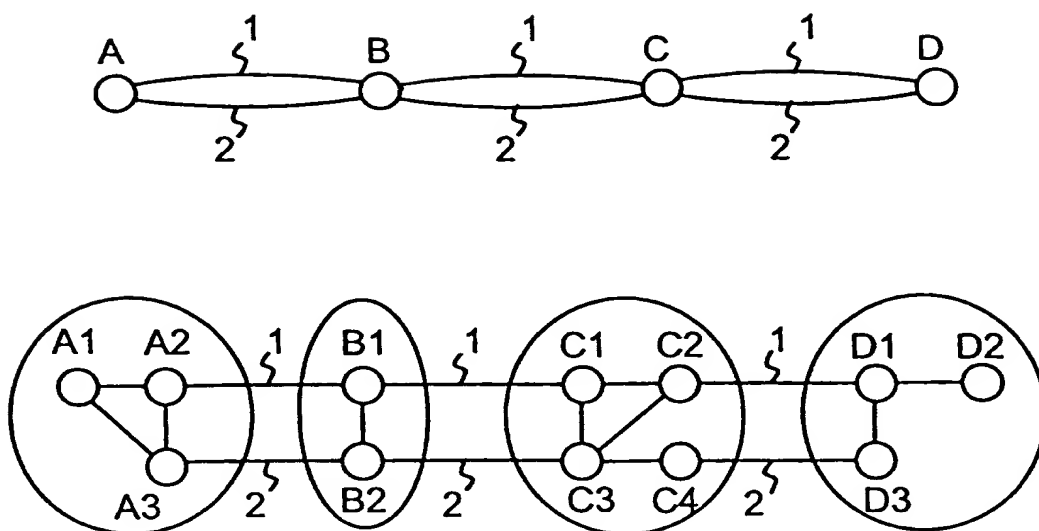


Figure 3

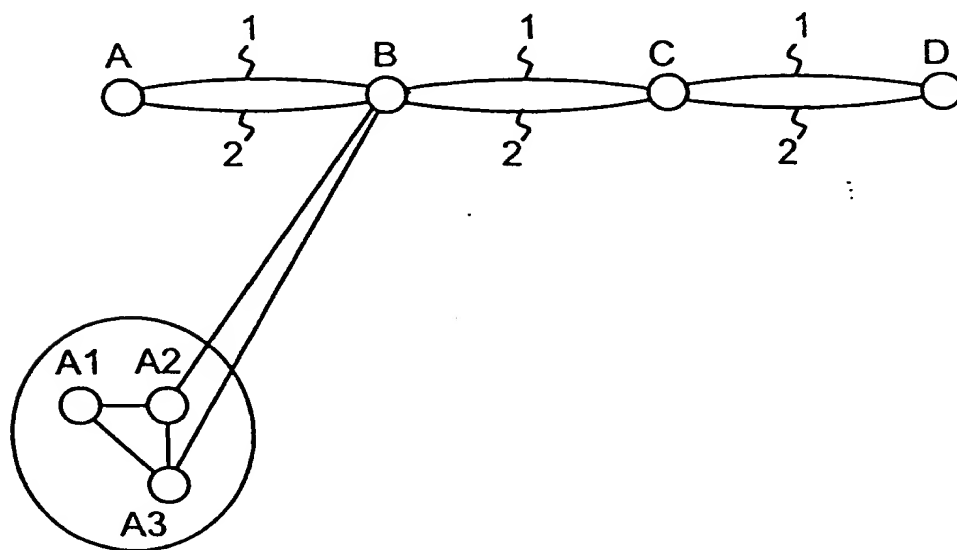


Figure 4

BEST AVAILABLE COPY